RT&A, No 1 (82)
Volume 20, March 2025

ENHANCING INTRUSION DETECTION SYSTEM RELIABILITY USING GWO-SOMNN (GREY WOLF OPTIMIZATION WITH SELF-ORGANIZING MAP NEURAL NETWORK)

Ms. Archana Gondalia[1] , Dr. Apurva Shah[2]

•

[1] Assistant Professor, Computer Engineering Department,
L.D. College of Engineering, Gujarat, India
archana.gondalia@gujgov.edu.in
[2] Professor, Department of Computer Science & Engineering,
The Maharaja Sayajirao University of Baroda, Gujarat, India
apurva.shah-cse@msubaroda.ac.in

Abstract

In today's fast-changing technological environment, the number of Internet-connected devices has grown significantly, raising the risk of cybersecurity threats for both individuals and organizations. Network Intrusion Detection Systems (NIDS) have become vital tools for protecting networks from these increasing threats. This paper presents a GWO-SOMNN approach (Gray Wolf Optimization with Self-Organizing Map Neural Network) that combines Grey Wolf Optimization (GWO), Self-Organizing Maps (SOM) and Neural Networks (NN) for feature selection and classification on the UNSW-NB15 dataset. The proposed method leverages GWO to optimize feature selection, reducing the dataset's dimensionality and computational load, while SOM is employed for clustering and visualizing high-dimensional data. Neural Networks are then used for effective classification of network attacks. The GWO-SOMNN approach is evaluated on the UNSW-NB15 dataset, and its performance is measured in terms of 97.18% accuracy and 97.15% F1-score for binary classification and 82.41% accuracy and 78.92% F1-score for multiclass classification. The results demonstrate significant improvements over traditional methods, particularly in enhancing the classification of both binary and multi-class network attacks. These findings highlight the potential of this integrated approach in developing more efficient and accurate network intrusion detection systems.

Keywords: Grey Wolf Optimization, Neural Networks, Self-Organizing Maps, Classification, Intrusion Detection, reliability

1. Introduction

In this section, a brief background introductory note relating to the development and evaluation of a hybrid approach combining Grey Wolf Optimization (GWO), Self-Organizing Maps (SOM), and Neural Networks (NN) for enhanced feature selection and classification in UNSWNB-15 datasets is presented in brief. In the rapidly evolving landscape of UNSWNB-15, the ability to efficiently and accurately detect threats and intrusions is paramount [1]. As cyber-attacks become more sophisticated, traditional methods of threat detection struggle to keep up. This research focuses on enhancing the detection and classification of cyber threats through the development and evaluation of a novel hybrid approach that integrates GWO, SOM, and NN. Feature selection

plays a critical role in UNSWNB-15, as the enormous volume of data produced by network systems can overcome traditional detection mechanisms [2]. By identifying the most relevant features, we can reduce the complexity and improve the performance of classification algorithms. GWO, a nature-inspired metaheuristic algorithm, offers a promising solution for optimal feature selection due to its simplicity and efficiency [3].

In the digital era, cybersecurity attacks have escalated in both frequency and sophistication, presenting substantial risks to individuals, businesses, and governments alike. This surge in cyber threats is closely linked to the increasing dependence on technology, the widespread adoption of Internet of Things (IoT) devices, and the exponential growth of online data. Cybercriminals exploit vulnerabilities in networks and systems, targeting sensitive information and launching attacks such as ransomware, data breaches, phishing, and distributed denial-of-service (DDoS). These incidents can occur unexpectedly, leading to operational disruptions, financial loss, and reputational damage.

An Intrusion Detection System (IDS) is a crucial cybersecurity tool that monitors network traffic and system activities for malicious activities or policy violations. It can be categorized into Network-based IDS (NIDS) and Host-based IDS (HIDS). IDS helps identify vulnerabilities within the network or system, providing alerts to administrators to mitigate risks. It also maintains the integrity and confidentiality of sensitive data, reducing the likelihood of data breaches. IDS is a crucial element of a multi-layered security strategy, working with firewalls, antivirus software, and other security measures to protect against evolving cyber threats. As technology evolves, new and advanced attacks emerge, exploiting weaknesses in hardware, software, and human behavior. Zero-day exploits, IoT devices, botnets, and remote work increase the attack surface. Advanced persistent threats (APTs) and ransomware attacks are also growing. Organizations must stay vigilant and proactive in their cybersecurity efforts, including regular software updates, multi-layered security strategies, and employee education on best practices.

The integration of GWO, SOM, and NN in our GWO-SOMNN approach offers several advantages. GWO ensures optimal feature selection, reducing dimensionality and computational complexity. SOM aids in clustering and visualizing the selected features, enhancing interpretability. NN provides robust classification, leveraging the refined feature set for accurate threat detection. This research contributes to the field of UNSWNB-15 by proposing a novel method for feature selection and classification. By combining the strengths of GWO, SOM, and NN, we aim to develop a solution that addresses the limitations of traditional methods and offers improved performance. The results of our evaluation demonstrate the potential of the hybrid approachto attain a high degree of categorization precision and provide a robust solution for network intrusion detection.

The remaining part of the paper is structured as follows: The prior research on anomaly detection with machine learning techniques is covered in Section 2. The data set is described in Section 3. The proposed hybrid approach presented in Section 4. After outlining the experimental parameters and performance metrics, Section 5 presents results and discussion of proposed GWO-SOMNN approach with the state-of-the-art methods, and Sections 6 and 7 present conclusions and future work plans, respectively.

2. Related Work

In the literature, numerous models for intrusion detection have been presented. This section covers a number of deep learning, machine learning, and data mining-based intrusion detection models.

In a novel approach, a weight embedding autoencoder was proposed by authors in [4] to enhance feature representation in network intrusion detection systems. This method facilitates the sharing of feature representations between the autoencoder and classifier, leading to improved detection accuracy. Their experiments on the NSL-KDD and UNSW-NB15 datasets demonstrate the model's effectiveness, with accuracy improvements of up to 2.8% on UNSW-NB15 and 0.5% on NSL-KDD. An IDS framework was implemented by by authors in [5], utilizing various

Recurrent Neural Networks (LSTM, GRU, and Simple RNN) to enhance network security. To improve detection accuracy, they applied an XGBoost-based feature selection algorithm on the NSL-KDD and UNSW-NB15 datasets. Their results indicate that XGBoost-LSTM achieved the best performance in binary classification, while XGBoost-GRU performed well for multiclass classification on these datasets.

An intrusion detection model was proposed by authors in [6], utilizing an Improved Social Network Search (ISNS) algorithm to optimize the BP neural network. By incorporating chaotic mapping and an elite mechanism into the original SNS algorithm, they successfully mitigated the BP network's tendency to get trapped in local optima. The optimized model, ISNS_BP, demonstrated superior classification accuracy on the NSL-KDD and UNSW-NB15 datasets, achieving 98.62% and 93.97%, respectively. A novel approach for uncertainty quantification in anomaly detection was introduced by authors in [7], using Bayesian Autoencoder (BAE) models. Their method incorporates heteroscedastic aleatoric uncertainty modeling, jointly accounting for both aleatoric and epistemic uncertainties. Applied to cybersecurity datasets such as UNSW-NB15 and CIC-IDS-2017, this framework enhances the trustworthiness of anomaly predictions, reducing false positives and improving decision-making in cybersecurity.

A hybrid method for anomaly detection in IoT devices, called CNN-BMECapSA-RF, was implemented by [8]. This approach combines a convolutional neural network (IoTFECNN) for feature extraction and a binary multi-objective Capuchin Search Algorithm (BMECapSA) for feature selection. Tested on the NSL-KDD and TON-IoT datasets, it achieved high accuracy rates of 99.99% and 99.85%, respectively, by identifying 27% and 44% of relevant features, outperforming existing deep learning and machine learning-based approaches. In another study, [9] proposed a novel intrusion detection approach, LR-ABC, which combines logistic regression (LR) with the artificial bee colony (ABC) algorithm for hyper-parameter optimization. The model improves the accuracy and reliability of network intrusion detection systems (NIDS) by addressing limitations in metrics such as accuracy, F1-measure, and false positives. Tested on the UNSW-NB15 and NSL-KDD datasets, the LR-ABC model achieved accuracy scores of 88.25% and 90.11%, respectively, demonstrating its effectiveness in enhancing detection systems.

Additionally, [10] introduced a hybrid Hunger Games Search and Remora Optimization Algorithm (HHGS-ROA) to tackle security issues in IoT networks. This model enhances the performance of intrusion detection systems by extracting relevant features from the Aegean Wi-Fi Intrusion Dataset (AWID) and classifying network traffic as either normal or malicious using an SVM classifier. The approach outperformed existing methods, achieving high accuracy (99.16%) and a low false-positive rate (0.20%), along with improved metrics such as precision, recall, and F1 score.

3. Dataset Description

The UNSW-NB15 dataset was utilized in this study to detect and classify network intrusions. It contains both normal and abnormal network traffic, with a total of nine categories representing different types of attacks alongside normal traffic. These categories include Denial of Service (DoS), Reconnaissance, Exploits, Backdoors, Fuzzers, Generic attacks, Analysis, Shellcode, and Worms, offering a diverse and comprehensive range of attack patterns for effective evaluation of intrusion detection systems (IDS). The distribution of training and testing set of UNSW-NB15 data set is shown in Table 1 [11].

Table 1: *Class distribution of UNSW-NB15*

Symbols		Set Size	
Type	Name	Training Set	Testing Set
0	Normal	56,000	37,000
1	Backdoor	1,746	583
2	Analysis	2,000	677
3	Fuzzers	18,184	6,062
4	Shellcode	1,133	378
5	Reconnaissance	10,491	3,496
6	Exploit	33,393	11,132
7	DoS	12,264	4,086
8	Worms	130	44
9	Generic	40,000	18,871

The dataset consists of 49 features that describe various aspects of the network traffic. These features were generated using twelve distinct algorithms applied to the raw traffic captured by the TCP dump tool. This diverse set of features makes the UNSW-NB15 dataset suitable for assessing the performance of machine learning and deep learning models in the field of intrusion detection. The dataset provides a challenging environment for anomaly detection, offering a balanced representation of modern attack types in network security research.

4. Proposed GWO-SOMNN Approach

In this section, we present the GWO, SOM and NN, a hybrid approach along with how do we evaluate the entire process which is shown in fig 1. The proposed GWO-SOMNN approach integrates GWO for feature selection and SOM combined with NN for classification. The GWO is employed to optimize feature subsets, enhancing the model's efficiency by selecting the most relevant features from the UNSW-NB15 dataset. Subsequently, SOM visualizes the data patterns while the Neural Network accurately classifies it into attack or normal categories, ensuring a robust intrusion detection mechanism. In this research, each component GWO, SOM, and NN plays a specific role in detecting and classifying intrusions. A detailed explanation of their roles is presented.

4.1. Grey Wolf Optimization

GWO is inspired by the hierarchical social structure and hunting behavior of grey wolves (*Canis lupus*). In this optimization algorithm, the population of candidate solutions is categorized into four main groups based on their leadership hierarchy: alpha (α), beta (β), delta (δ), and omega (ω).

- **Alpha wolves (α)** are considered the most dominant and lead the pack. They are responsible for decision-making and guiding the hunting process.
- **Beta wolves (β)** hold the second rank and assist the alpha in decision-making while also enforcing the alpha™s commands within the pack.
- **Delta wolves (δ)** are subordinate to both alpha and beta but rank higher than omega wolves. This group includes hunters, scouts, and sentinels. Hunters are responsible for locating prey and providing food for the pack. Scouts monitor the surroundings for threats, while sentinels ensure the pack's safety.

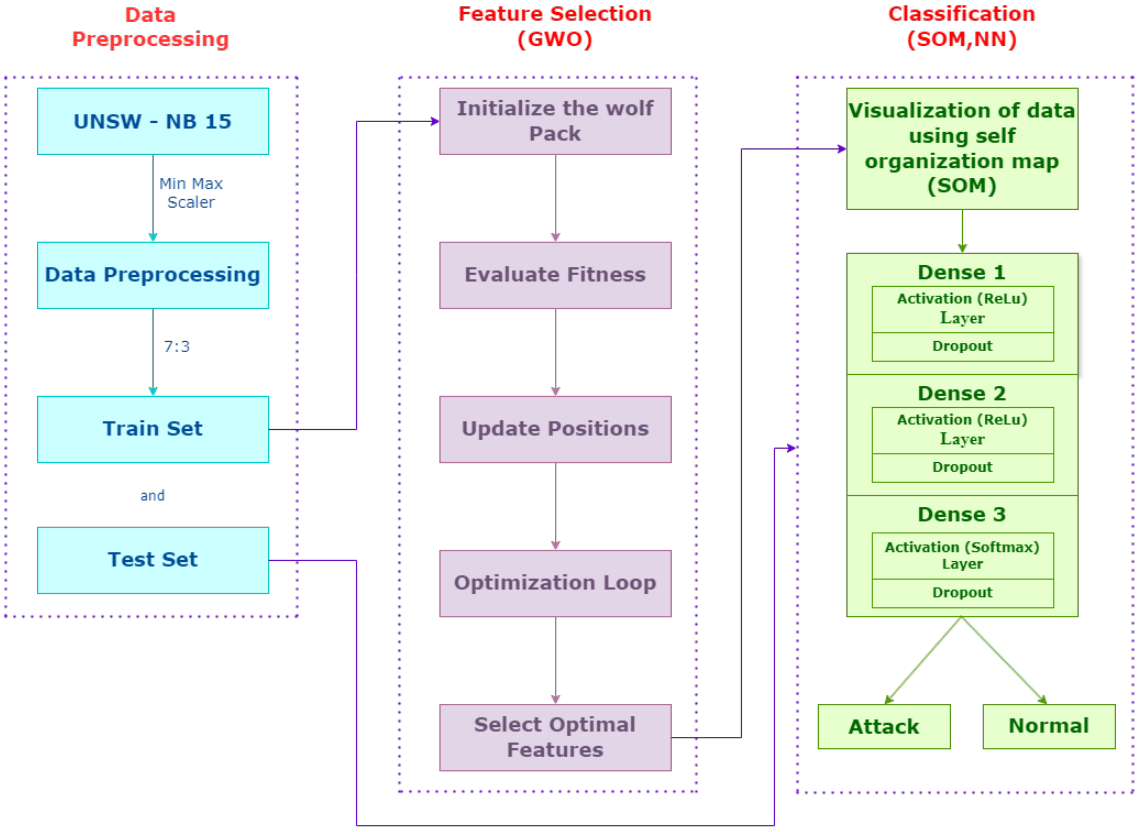

Figure 1: *GWO-SOMNN Approach for IDS*

- **Omega wolves (ω)** are the lowest in the hierarchy. They play a crucial role in maintaining pack structure by following orders from the other groups, especially during hunting activities.

The hierarchical structure ensures that information from the environment is processed and actions are taken efficiently, allowing the wolves to hunt successfully.

GWO mimics this natural hunting mechanism, where the wolves encircle their prey during the hunt. The position of the prey (optimal solution) is estimated by the leading wolves (α, β, and δ), while the remaining wolves update their positions relative to these leaders. The behavior of encircling prey can be mathematically modeled using the following equations [12]:

$$\vec{F}(t+1) = \vec{F}_p(t) - \vec{A} \cdot \vec{D} \tag{1}$$

$$\vec{D} = \left| \vec{C} \cdot \vec{F}_p(t) - \vec{F}(t) \right| \tag{2}$$

Where $\vec{F}_p(t)$ represents the position of the prey, $\vec{F}(t)$ is the position of a grey wolf, and $\vec{A}$ and $\vec{C}$ are coefficient vectors used to simulate the encircling behavior. These vectors are calculated as follows:

$$\vec{a} = 2 - t \left(\frac{2}{\text{Max}_{\text{iter}}} \right) \tag{3}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{4}$$

$$\vec{C} = 2\vec{r}^2 \tag{5}$$

Where t is the current iteration, Max_{iter} is the maximum number of iterations, and $\vec{r}_1$ and $\vec{r}_2$ are random vectors in [0,1].

The algorithm proceeds by iteratively updating the positions of the wolves, with the alpha, beta, and delta wolves guiding the optimization process. Over time, the wolves converge toward the optimal solution, mimicking the grey wolves' real-life hunting strategy.

GWO algorithm, the natural hunting strategy of grey wolves is emulated to optimize search processes. Grey wolves typically locate and encircle their prey, led by the alpha wolf, with the beta and delta occasionally assisting, adding a layer of complexity as the prey™s exact location within the search space is often unknown. In GWO, this behavior is simulated by treating the alpha, beta, and delta wolves as having the most accurate knowledge of the prey's whereabouts, making them the primary guides in the search. The remaining wolves, including omegas, adjust their positions based on the guidance from these top three wolves. The algorithm keeps these top three candidate solutions at the forefront of the process and dynamically adjusts the positions of all other wolves through a following set of equations, effectively simulating the encircling and attacking phases of wolf hunting [13].

$$\vec{D}_\alpha = \left| \vec{D}_1 \cdot \vec{F}_\alpha - \vec{F} \right| \tag{6}$$

$$\vec{D}_\beta = \left| \vec{D}_2 \cdot \vec{F}_\beta - \vec{F} \right| \tag{7}$$

$$\vec{D}_\delta = \left| \vec{D}_3 \cdot \vec{F}_\delta - \vec{F} \right| \tag{8}$$

$$\vec{F}_1 = \vec{F}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \tag{9}$$

$$\vec{F}_2 = \vec{F}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \tag{10}$$

$$\vec{F}_3 = \vec{F}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \tag{11}$$

$$\vec{F}(t+1) = \frac{\vec{F}_1 + \vec{F}_2 + \vec{F}_3}{3} \tag{12}$$

In the GWO algorithm, the equations above describe the process by which the positions of grey wolves (potential solutions) are updated based on the positions of the three leading wolves"alpha ($\vec{F}_\alpha$), beta ($\vec{F}_\beta$), and delta ($\vec{F}_\delta$). In this model, the grey wolves encircle their prey, represented by the optimal solution.

The first set of equations calculates the distance vectors ($\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$) between the current wolf's position ($\vec{F}$) and each of the leading wolves ($\vec{F}_\alpha$, $\vec{F}_\beta$, and $\vec{F}_\delta$), adjusted by dynamic coefficients ($\vec{D}_1$, $\vec{D}_2$, and $\vec{D}_3$) to control the movement towards these leaders.

Subsequently, the positions of the wolves ($\vec{F}_1$, $\vec{F}_2$, and $\vec{F}_3$) are updated by subtracting a second set of coefficients ($\vec{A}_1$, $\vec{A}_2$, and $\vec{A}_3$) scaled by the distance vectors. Finally, the new position of each grey wolf ($\vec{F}(t+1)$) is calculated as the average of the positions derived from the three leaders, ensuring that the wolves converge towards the prey, which represents the optimal solution in the search space. This process is repeated iteratively until convergence is achieved.

In this implementation of the GWO algorithm for feature selection, two primary parameters are adjusted: 'SearchAgents_no' and 'Max_iter'. The 'SearchAgents_no' is set to 5, indicating the number of grey wolves (agents) used to explore the search space, which directly affects the diversity of potential solutions. The 'Max_iter' parameter is set to 100, controlling the maximum number of iterations for the optimization process, ensuring a balance between computational cost and optimization depth. Additionally, upper ('ub') and lower ('lb') bounds for the feature selection space are defined, allowing features to be represented as binary values (0 or 1). The exploration-exploitation balance is controlled through the 'a' parameter, which linearly decreases over iterations, guiding the wolves' movements from global exploration to local exploitation. Random vectors 'r1' and 'r2' introduce variability, making the search process robust by allowing each wolf to update its position relative to the best (Alpha), second-best (Beta), and third-best

Table 2: *The list of features selected by GWO*

Feature Information		
Sr. No	**Feature Number**	**Feature Name**
1	1	dur
2	2	proto
3	3	service
4	7	sbytes
5	8	dbytes
6	9	rate
7	11	dttl
8	12	sload
9	17	dinpkt
10	20	swin
11	25	synack
12	27	smean
13	28	dmean
14	30	response_body_len
15	33	ct_dst_ltm
16	34	ct_src_dport_ltm
17	35	ct_dst_sport_ltm
18	38	ct_ftp_cmd
19	43	label

(Delta) solutions. This configuration ensures that the algorithm efficiently searches for the most optimal subset of features.

GWO selected key features in Table 2 based on network flow characteristics for enhancing classification performance. These features include attributes like connection duration, protocol types, data rate, and packet statistics. Features such as source-to-destination transaction bytes, TCP window advertisement, and SYN-ACK flags are critical in identifying attack patterns. This optimized feature subset enables a more efficient and accurate detection of network intrusions.

4.2. Self-Organizing Maps

SOMs are widely used for grouping and displaying high-dimensional data in a lower-dimensional area [14].

The SOM algorithm begins with the initialization of a weight matrix, which is randomly assigned and represents the position of each neuron within the input feature space. The algorithm iteratively maps data points to the SOM grid, identifying the "best matching unit" (BMU), or winner neuron, for each input. The weights of the BMU and its neighboring neurons are then adjusted, bringing them closer to the input data point. This iterative process allows the SOM to progressively refine its mapping and organization of the data. SOMs are particularly advantageous in exploratory data analysis, offering researchers the capability to uncover latent patterns or groupings within complex datasets. Additionally, SOMs serve as a robust tool for data visualization, enhancing the interpretability and analysis of data across various fields, including bioinformatics, finance, and marketing.

Initialization: For each input vector $\vec{W}_{(i,j)}$ for each neuron (i, j)
Training: For each input vector $\vec{x}$:
Find Best Matching Unit (BMU): Here, we use the BMU model as:

$$\text{BMU} = \arg\min_{i,j} \|\vec{x} - \vec{W}_{i,j}\| \tag{13}$$

$$\vec{W}_{i,j}(t+1) = \vec{W}_{i,j}(t) + \theta(t,i,j) \cdot \alpha(t) \cdot (\vec{x} - \vec{W}_{i,j}(t)) \tag{14}$$

where, $\theta(t, i, j)$ is the neighborhood function, which decreases over time. $\alpha(t)$ is the learning rate, which also decreases over time.

After selecting the optimal features using the GWO algorithm, a SOM is employed to visualize and further process the selected data. In this implementation, the SOM is initialized with a 5x5 grid ('x=5, y=5') to create a map of neurons that represents the input space. The 'input_len' parameter is dynamically set to match the number of features selected by GWO, ensuring that each neuron can accommodate the reduced feature set. The 'sigma' parameter, which controls the radius of influence for each neuron during the learning process, is set to 1.0, allowing a moderate neighborhood influence on the weight updates. The learning rate is initialized at 0.5, guiding the network's convergence speed as it adapts to the data. The SOM is trained using random samples from the training set for 25 iterations, facilitating the clustering and visualization of attack and normal data in an unsupervised manner. This approach allows the model to discover inherent patterns in the dataset and enhances its ability to differentiate between attack and normal classes.

4.3. Neural Networks

NNs represent a fundamental element of contemporary artificial intelligence, drawing inspiration from the structure and function of the human brain. These networks consist of multiple layers of interconnected neurons that process input data, enabling the system to learn and recognize patterns[15].

We have used the Multilayer Perceptron (MLP) neural network in our work, which is a fundamental type of artificial neural network, characterized by its feedforward architecture, where data flows in one direction"from the input layer through one or more hidden layers to the output layer. This structure makes MLPs particularly effective for supervised learning tasks, such as classification and regression. The MLP begins with an input layer, which serves as the entry point for the data. Each neuron in this layer corresponds to a specific feature of the input data. In our data set out of 45 features, the GWO algorithm has selected 19 features for binary and multiclass classification, so the input layer will have 19 neurons, each representing one of those features. Following the input layer with one or more hidden layers. These layers are the core of the MLP, where the actual computation and learning take place. Each neuron in a hidden layer is connected to every neuron in the previous layer, forming a fully connected network. The neurons in the hidden layers perform computations by applying a weighted sum of the inputs from the previous layer, followed by an activation function ReLU (Rectified Linear Unit), which introduces non-linearity to the model. The Fig. 2 and 3 gives the NN diagram for binary classification and multiclass classification respectively.

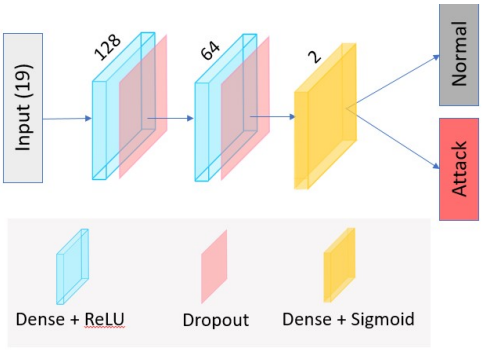

Figure 2: *Multilayer Perceptron Neural network diagram for binary classification*

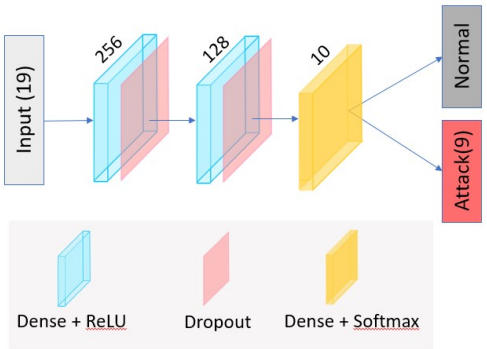

Figure 3: *Multilayer Perceptron Neural network diagram for multi class classification*

The final layer of the MLP is the output layer, which produces the model's predictions or classifications based on the processed data. The number of neurons in the output layer corresponds to the number of possible outputs. For binary classification, the output layer has

two neuron with a Sigmoid activation function to produce a probability score and multi-class classification has also used softmax activation function with multiple neurons, one for each class. The forward propagation process in an MLP involves passing the input data through the network, layer by layer. Each neuron computes a weighted sum of its inputs, applies the activation function, and passes the result to the next layer, culminating in the output layer's final prediction.

The neural network model is constructed using TensorFlow™s Keras library and consists of three layers, three dense (fully connected) layers. The number of parameters in the neural network depends on the sizes of these layers and the input size. We chose the size of the neural network (128, 64, 2) is shown in Table 3. The first dense layer has 128 neurons, with parameters calculated as 128 — input_len™ + 128 (weights plus biases). The second dense layer has 64 neurons, with parameters 128 — 64 + 64 = 8,256. The output layer, with 2 neurons for binary classification, has parameters 64 — 2 + 2 = 130. Thus, the total parameters in the neural network are 128 — input_len™ + 8,514, where input_len™ describes the number of selected features with GWO algorithm. Here, 19 optimal features are selected by applying the GWO algorithm, so 10,946 total parameters are used in the neural network.

Table 3: *Model Architecture and Parameters*

Layer Type	Output Shape	Activation	Parameters
Input Layer	19	-	0
Dense	128	ReLU	3,072
Dropout	0.5	-	-
Dense	64	ReLU	8,256
Dropout	0.5	-	-
Dense	2	Sigmoid	130
Total Parameters			**10,946**

Table 4: *Model Architecture and Parameters*

Layer Type	Output Shape	Activation	Parameters
Input Layer	19	-	0
Dense	256	ReLU	4,352
Dropout	0.5	-	0
Dense	128	ReLU	32,896
Dropout	0.5	-	0
Dense	10	Softmax	1,290
Total Parameters			**39,306**

Similary, for multi-class classification, we chose neural network size (256, 128, 10) is shown in table 4. The first dense layer has 256 neurons, with parameters calculated as 256 — input_len™ + 256 (weights plus biases). The second dense layer has 128 neurons, with parameters 256 — 128 + 128 = 32,896. The output layer, with 10 neurons for muticlass classification, has parameters 128 — 10 + 10 = 1290. Thus, the total parameters in the neural network are 256 — input_len™ + 34,442, where input_len™ describes the number of selected features with GWO algorithm. Here, 19 optimal features selected by applying GWO algorithm so 39,306 total parameters used in neural network.

5. Results and Discussion

In this research, the proposed method implemented in the Google Colab environment, the developed code was run & the simulation results were observed and displayed for UNSWNB15 Binary Classification as well as for Multiclass Classifications, the results are specified separately.

5.1. Evaluation Metrics

The following metrics are used to assess the proposed hybrid approach: accuracy, precision, recall and F-measure. The following defines each measure:

- - Accuracy represents the proportion of correctly classified records out of the total dataset.
- Precision refers to the percentage of correctly identified anomalies among all records predicted to be anomalies.
- Recall, also known as the True Positive Rate or detection rate, is the percentage of actual anomalies that were correctly classified.
- F-measure is a metric that balances both precision and recall, providing a single performance measure.

5.2. Results

The experiment was conducted using the UNSW-NB15 dataset, with the GWO algorithm applied to select optimal features, SOM for data visualization, and MLP NN for classification. In Tables 5, the following abbreviations are used: NU (Number of hidden Units), TAC (Training Accuracy), VAC (Validation Accuracy), ET (Execution Time), and TEC (Testing Accuracy).

The experiment was performed in two phases. In the first phase, for binary classification, the GWO algorithm selected various features based on the SOM grid size of 5x5, adjusted according to the 3 hidden units. The activation functions used for the dense layers were ReLU, ReLU and Sigmoid. Additionally, the training time (in seconds) was recorded for each model.

The following hyperparameters were used for binary classification:

- Loss function: 'binary_crossentropy'
- Optimizer: 'adam' (an extension of Stochastic Gradient Descent)

In the second phase, for multiclass classification, the GWO algorithm selected different features, and a SOM grid size of 5x5 was used based on varying hidden units. The following hyperparameters were used for Multiclass Classification:

- loss ='sparse_categorical_crossentropy'
- optimizer = 'adam'' (an extension of Stochastic Gradient Descent).

Table 5: *Performance metrics of the proposed GWO-SOMNN approach for Binary and Multiclass Classification*

Classification	NU	TAC (%)	VAC (%)	ET (Sec)	Testing (%)	Precision (%)	Recall (%)	F1-Score (%)
Binary	128,64,2	97.24	97.18	466.29	97.18	97.24	97.18	97.15
Multiclass	256,128,2	82.41	82.33	43.247	82.33	76.60	82.33	78.92

Table 5 presents the performance metrics of the proposed Grey Wolf Optimization combined with Self-Organizing Map and Neural Network (GWO-SOMNN) approach for both binary and multiclass classification tasks. For binary classification, the model achieved a TAC of 97.24%, VAC of 97.18%, and required 466.29 seconds of execution time. The Testing Accuracy, Precision, Recall,

and F1-Score for binary classification were 97.18%, 97.24%, 97.18%, and 97.15%, respectively. For multiclass classification, the model achieved a TAC of 82.41%, VAC of 82.33%, and the execution time was 43.247 seconds. The Testing Accuracy, Precision, Recall, and F1-Score for multiclass classification were 82.33%, 76.60%, 82.33%, and 78.92%, respectively. These results indicate that the GWO-SOMNN approach performs better in binary classification compared to multiclass classification, especially in terms of accuracy and precision. Evaluation measures for Binary and multiclass classification is presented in Figure 4 and 5 respectively.

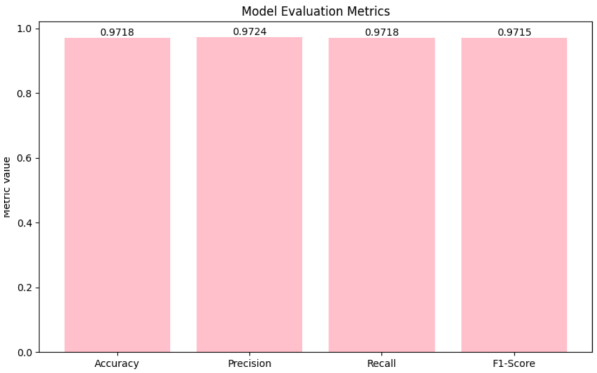

Figure 4: *Evaluation measures of Binary Classification*

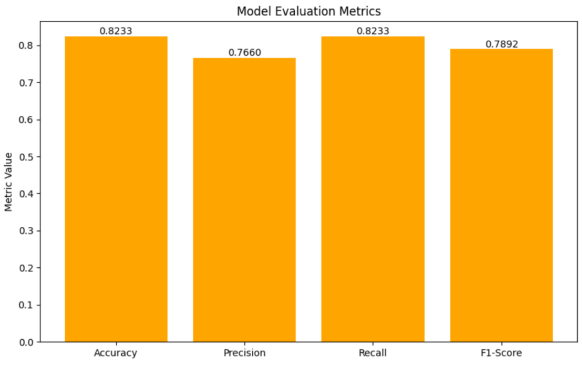

Figure 5: *Evaluation measures of Multiclass Classification*

5.3. Confusion Matrix

For each model, a Confusion Matrix (CM) was generated to evaluate the model's performance on individual classes within the datasets. In Figure 6, class 0 represents normal traffic, and class 1 represents attacks. In Fig. 5, the classes are defined as: class 0 = Normal, class 1 = Generic, class 2 = Exploits, class 3 = Fuzzers, class 4 = DoS, class 5 = Reconnaissance, class 6 = Analysis, class 7 = Backdoor, class 8 = Shellcode, and class 9 = Worms.

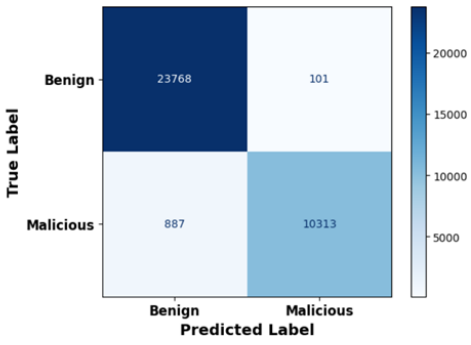

Figure 6: *Binary Class Confusion Matrix*

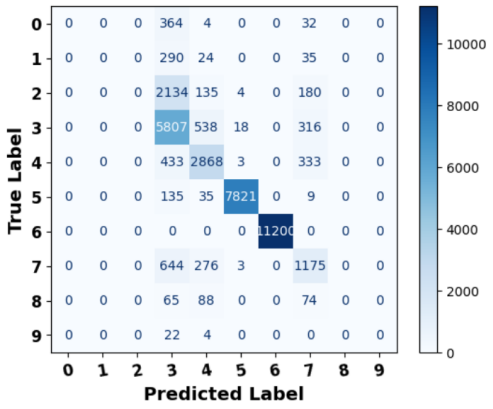

Figure 7: *Multiclass Confusion Matrix*

The GWO-SOMNN approach in Figure 6 correctly classified 23,768 benign instances and 10,313 malicious instances. However, some false negatives were observed, with 887 benign instances misclassified as malicious, alongside 101 false positives, where malicious instances were incorrectly classified as benign. In Figure 7, the confusion matrix reflects the GWO-SOMNN approach's performance across individual classes of the UNSW-NB15 dataset. The multiclass confusion matrix shows varying performance across different classes. Based on the confusion matrix, the approach correctly predicted classes such as Fuzzers (label 6), which demonstrated high precision with minimal misclassifications. Similarly, Normal traffic (label 5) was also

predicted with strong accuracy, as evidenced by the dense diagonal and almost no off-diagonal entries.

However, certain attack types, such as Generic (label 3) and Exploits (label 2), were more challenging for the model to classify accurately. The confusion matrix shows considerable misclassification for these attacks, where instances of Generic were confused with Exploits, and vice versa. This misclassification can be attributed to the overlapping feature patterns between these attack types, as both may share similar network characteristics, making it harder for the model to differentiate them effectively. Moreover, Reconnaissance (label 4) also exhibited some degree of misclassification, potentially due to its similarity in network behavior to other less specific attack types.

5.4. Discussion

For the binary classification, Table 6 compares the proposed methodology (PM) with other research approaches. The proposed GWO-SOMNN approach achieved a significant accuracy of 97.65% using 19 selected features. In contrast, previous methods such as Weight Embedding AutoEncoder with Convolutional Neural Network (CNN) and Multi-Layer Neural Network (MLNN) showed lower accuracies of 79.49% and 80.39%, respectively with 42 features. Other approaches like XGBoost variants (LSTM, GRU, Simple-RNN) demonstrated accuracies ranging from 85.08% to 87.07% with 17 features, whereas the Multivariate Correlations Analysis with Long Short-Term Memory (MCA-LSTM) with 14 features and Feed-Forward Deep Neural Network (FFDN) with 18 features achieved accuracies of 88.1% and 87.1%, respectively. The proposed approach clearly outperforms these methods in terms of accuracy for binary classification.

Table 6: *Comparison of Binary Classification*

Author	Method	Accuracy (%)
[16]	Weight Embedding AutoEncoder with a Convolutional Neural Network (WE-AE CNN)	79.49
[16]	Weight Embedding AutoEncoder with a Multi-Layer Neural Network (WE-AE DNN)	80.39
[17]	XGBoost-LSTM	85.08
[17]	XGBoost-GRU	88.42
[17]	XGBoost-Simple-RNN	87.07
[18]	Multivariate Correlation Analysis “ Long Short-Term Memory Network (MCA-LSTM)	88.1
[19]	Feed-Forward Deep Neural Network (FFDN)	87.1
PM	**GWO-SOMNN Approach**	**97.65**

For multiclass classification, as shown in Table 7, the proposed hybrid approach also outperforms other methods, achieving an accuracy of 82.41% with 19 selected features. In comparison, Weight Embedding AutoEncoder with CNN and MLNN reached accuracies of 74.19% and 73.01%, respectively with 42 features, while XGBoost-GRU with 17 features and FFDN with 18 features yielded accuracies of 78.4% and 77.16%. Thus, the proposed GWO-SOMNN approach shows an improvement in multiclass classification accuracy, demonstrating its effectiveness in handling complex, real-world datasets.

6. Conclusion

This research introduced a hybrid approach integrating Grey Wolf Optimization (GWO), Self-Organizing Maps (SOM), and Neural Networks (NN) to enhance feature selection and classification for intrusion detection using the UNSW-NB15 dataset. The main goal was to optimize feature selection and clustering to improve the performance of intrusion detection systems. By leveraging

Table 7: *Comparison of Multiclass Classification*

Author	Method	Accuracy (%)
[16]	Weight Embedding AutoEncoder with a Convolutional Neural Network (WE-AE CNN)	74.19
[16]	Weight Embedding AutoEncoder with a Multi-Layer Neural Network (WE-AE DNN)	73.01
[17]	XGBoost-GRU	78.4
[19]	Feed-Forward Deep Neural Network (FFDN)	77.16
PM	**GWO-SOMNN Approach**	**81.53**

GWO for efficient feature selection and SOM for data visualization, the GWO-SOMNN approach significantly reduced the dataset's dimensionality, improving the computational efficiency and accuracy of neural network-based classification.

The results demonstrate that the proposed method outperforms traditional techniques in both binary and multiclass classifications, achieving notable improvements in accuracy, precision, recall, and F1-score. Specifically, the GWO-SOMNN approach achieved a binary classification accuracy of 97.18% and a multiclass classification accuracy of 82.41%, surpassing many state-of-the-art methods. This indicates the potential of this integrated approach for developing more efficient and precise network intrusion detection systems.

7. Future Work

Future work will focus on the real-time implementation and scalability of the GWO-SOMNN approach. Deploying this model in live network environments will allow for the evaluation of its performance under real-time conditions. Additionally, extending the model to handle larger and more complex datasets will test its robustness and scalability in diverse scenarios.

Further improvements could include optimizing the GWO algorithm by integrating it with other metaheuristic techniques such as Particle Swarm Optimization or Genetic Algorithms. Also, incorporating advanced neural network architectures, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), may enhance feature selection efficiency and classification accuracy.

Exploring adversarial training techniques will help improve the model™s resilience against adversarial attacks, while the application of different SOM variants could enhance clustering and visualization. Finally, implementing automated hyperparameter tuning and incorporating behavioral analysis could further enhance the adaptability and detection of sophisticated threats.

References

[1] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.

[2] E. Emary, Hossam M. Zawbaa, and Aboul Ella Hassanien. Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172:371–381, 2016.

[3] nal §avuolu. A new hybrid approach for intrusion detection using machine learning methods. *Applied Intelligence*, 49, 07 2019.

[4] Mulyanto Mulyanto, Jenq-Shiou Leu, Muhamad Faisal, and Wawan Yunanto. Weight embedding autoencoder as feature representation learning in an intrusion detection systems. *Computers and Electrical Engineering*, 111:108949, 2023.

[5] Sydney Mambwe Kasongo. A deep learning technique for intrusion detection system using a recurrent neural networks based framework. *Computer Communications*, 199:113–125, 2023.

[6] Zhongjun Yang, Qi Wang, Xuejun Zong, and Guogang Wang. Intrusion detection method based on improved social network search algorithm. *Computers Security*, 140:103781, 2024.

[7] Tengfei Yang, Yuansong Qiao, and Brian Lee. Towards trustworthy cybersecurity operations using bayesian deep learning to improve uncertainty quantification of anomaly detection. *Computers Security*, 144:103909, 2024.

[8] Hossein Asgharzadeh, Ali Ghaffari, Mohammad Masdari, and Farhad Soleimanian Gharehchopogh. Anomaly-based intrusion detection system in the internet of things using a convolutional neural network and multi-objective enhanced capuchin search algorithm. *Journal of Parallel and Distributed Computing*, 175:1–21, 2023.

[9] Burak Kolukisa, Bilge Kagan Dedeturk, Hilal Hacilar, and Vehbi Cagri Gungor. An efficient network intrusion detection approach based on logistic regression model and parallel artificial bee colony algorithm. *Computer Standards Interfaces*, 89:103808, 2024.

[10] Ravinder Kumar, Amita Malik, and Virender Ranga. An intellectual intrusion detection system using hybrid hunger games search and remora optimization algorithm for iot wireless networks. *Knowledge-Based Systems*, 256:109762, 2022.

[11] Australian Centre for Cyber Security. Unsw-nb15 dataset, 2015. Accessed: 2024-09-16.

[12] Mohammad H. Nadimi-Shahraki, Hoda Zamani, Zahra Asghari Varzaneh, Ali Safaa Sadiq, and Seyedali Mirjalili. A systematic review of applying grey wolf optimizer, its variants, and its developments in different internet of things applications. *Internet of Things*, 26:101135, 2024.

[13] Reem Alkanhel, Doaa Sami Khafaga, El-Sayed M. El-kenawy, Abdelaziz A. Abdelhamid, Abdelhameed Ibrahim, Rashid Amin, Mostafa Abotaleb, and B. M. El-den. Hybrid grey wolf and dipper throated optimization in network intrusion detection systems. *Computers, Materials and Continua*, 74(2):2695–2709, 2022.

[14] Yang Chen, Nami Ashizawa, Chai Kiat Yeo, Naoto Yanai, and Seanglidet Yean. Multi-scale self-organizing map assisted deep autoencoding gaussian mixture model for unsupervised intrusion detection. *Knowledge-Based Systems*, 224:107086, 2021.

[15] Mouaad Mohy-eddine, Azidine Guezzaz, Said Benkirane, and Mourade Azrour. An intrusion detection model using election-based feature selection and k-nn. *Microprocessors and Microsystems*, page 104966, 2023.

[16] Mulyanto Mulyanto, Jenq-Shiou Leu, Muhamad Faisal, and Wawan Yunanto. Weight embedding autoencoder as feature representation learning in an intrusion detection systems. *Computers and Electrical Engineering*, 111:108949, 2023.

[17] Sydney Mambwe Kasongo. A deep learning technique for intrusion detection system using a recurrent neural networks based framework. *Computer Communications*, 199:113–125, 2023.

[18] Rui-Hong Dong, Xue-Yong Li, Qiu-Yu Zhang, and Hui Yuan. Network intrusion detection model based on multivariate correlation analysis " long short-time memory network. *IET Information Security*, 14(2):166–174, 2020.

[19] Sydney Mambwe Kasongo and Yanxia Sun. A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Computers Security*, 92:101752, 2020.